REMARKS

This Amendment is filed in response to the non-final Office Action mailed on December 2, 2004. All objections and rejections are respectfully traversed. Reconsideration of the application, as amended, is respectfully requested.

Claims 1-37 are pending.

Claims 20, 26, 28, 31 and 37 have been amended to better claim the invention.

Claims 38-45 have been added. Support for the newly added claims may be found, among other places, in Applicants' Fig. 6 and its related description. No new matter is being introduced.

The Applicants have amended page 12 of the specification to identify that U.S. Patent Application Serial No. 10/027,634 has been assigned to the patent application entitled FILE SYSTEM DEFRAGMENTATION TECHNIQUE VIA WRITE ALLOCATION, by John Edwards et al., filed December 21, 2001, which application was incorporated by reference in the present case.

On page 2 of the Office action, claim 37 was objected to because claims 10 and 26 should be recited in the alternative. In response, the Applicants have amended claim 37 to recite "the method of claim 10 **or** 26," as suggested. Accordingly, the Applicants submit that the objection to claim 37 should be removed.

At paragraphs 1-2 of the Office Action claims 1-4, 8-24 and 26-37 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,440,726 issued on August 8, 1995 to Fuchs et al. (hereinafter "Fuchs").

The present invention, as set forth in representative claim 1 comprises in part:

> 1. (Previously presented) A system for replay of a backup memory in a storage system having a file system for managing transfer of data to and from ***an attached disk array***, the system comprising:
>
> a log in the backup memory containing the storage system transaction entries accumulated after ***a consistency point at which time results of the storage system transaction entries are committed to the disk array***;
>
> ***an initiator process that establishes a swarm of messages with respect to the storage system transaction entries and delivers the swarm to the file system; and***
>
> ***a disk information-retrieval process in the file system that is carried out on the swarm of messages in parallel.***

Fuchs teaches a computing system that concurrently executes a plurality of different application processes. *See* col. 5, line 66 through col. 6, line 4. The processes communicate with one another by passing messages. *See* col. 2, lines 38-42. Each application process is associated with a corresponding nonvolatile (backup) memory 44 containing logs of the process's incoming and outgoing messages. *See* col. 6, lines 12-15 and fig. 1. The nonvolatile memory also stores the process's "critical" program data, which is transferred to the memory at regular "checkpoint" time intervals. *See* col. 3, lines 2-5.

In the event that an application process fails, a progressive retry recovery algorithm is performed. *See* col. 16, lines 22-27. The recovery algorithm replays each of the failed process's logged messages, one at a time, until the process's software fault has been by-passed. *See* fig. 14, refs. 1415-1445 and col. 1, lines 28-33.

The Applicants urge that Fuchs is legally precluded from anticipating or rendering obvious the Applicants' claim 1 because of its complete absence of ***a consistency point at which time results of the storage system transaction entries are committed to the disk array***, ***an initiator process that establishes a swarm of messages with respect to the storage system transaction entries and delivers the swarm to the file system***, and ***a disk information-retrieval process in the file system that is carried out on the swarm of messages in parallel***.

A. Fuchs does not teach or suggest an attached disk array, and thus cannot anticipate or render obvious the Applicants' claimed consistency point.

As shown in fig. 1 in Fuchs, each processing node includes a main memory unit and at least one processor that executes a corresponding application process P. Each application process P executing in the node is associated with a different set of volatile and nonvolatile memory areas in the main memory unit (i.e., P_0, P_1, ... P_N each have separately allocated memory areas). *See* col. 6, lines 12-15. No other memory units are disclosed in Fuchs.

Fuchs fails to disclose a system having ***an attached disk array***, as recited in the Applicants' claim 1. More specifically, the only memory elements disclosed in Fuchs are the processing nodes' volatile and nonvolatile memory areas, neither of which is an attached disk array as explicitly claimed. In fact, the words "disk" and "array" do not appear at all in the Fuchs reference.

Moreover, the Applicants' claim 1 explicitly recites two different types of memory elements: "an attached disk array" and "a backup memory." The Office action equates the Applicants' claimed backup memory with the nonvolatile memory in Fuchs. *See* Office action, para. 2. Accordingly, the only other memory element in Fuchs that can be equated with the Applicants' claimed disk array is Fuchs's volatile memory. However, Fuchs's volatile memory is "an area of unstable memory that is unable to retain information without continuous power" (Fuchs, col. 6, lines 16-18), and thus cannot read on the Applicants' claimed disk array.

Because of the complete absence of ***an attached disk array*** in Fuchs, Fuchs cannot teach or suggest ***a consistency point at which time results of the storage system transaction entries are committed to the disk array***, as claimed.

Here, it is further noted that Fuchs's "checkpoint" is not analogous to the Applicants' claimed "consistency point." Namely, the checkpoint in Fuchs specifies a time at which an application process's critical data is copied into the process's associated nonvolatile memory (*see* col. 3, lines 2-5), whereas the Applicants' consistency point is a time at which results of storage system transaction entries are committed to a disk array.

B. Fuchs does not teach or suggest grouping messages into a swarm of messages and delivering the swarm to a file system, as claimed.

At paragraph 2 in the Office action, the Applicant's claimed ***swarm of messages*** was rejected as being anticipated by the abstract in Fuchs. The Applicants have carefully

reviewed Fuchs's abstract, and respectfully submit that it does not teach or suggest ***an initiator process that establishes a swarm of messages with respect to the storage system transaction entries and delivers the swarm to the file system***, as recited in Applicants' claim 1.

Fuchs's abstract refers to message processing, as follows: (1) "A progressive retry recovery system based on checkpointing, message logging, rollback, message replaying and message reordering is disclosed" and (2) "functions are provided which process messages that are sent or received by an application process and maintain logs of the sent and received messages."

Neither of these message-passing references described in Fuchs's abstract teaches or suggests grouping multiple messages into a swarm (or any other unit, for that matter), nor do they teach or suggest delivering a swarm of messages to a file system. In fact, Fuchs expressly teaches that his message logging, replaying and reordering operations are performed one message at a time, and are not processed in groups (or "swarms") of messages. *See, e.g.,* fig. 14, refs. 1410-1445 (replaying messages one message at a time); col. 3, lines 33-34 (logging messages one message at a time, "read data from a communication channel and log the received message in the receiver log file").

Accordingly, because Fuchs fails to disclose processing swarms of messages and delivering the swarms to a file system, Fuchs cannot anticipate or render obvious Applicants' claimed ***initiator process that establishes a swarm of messages with respect to the storage system transaction entries and delivers the swarm to the file system.***

C. Fuchs does not teach or suggest processing a swarm of messages in parallel, as claimed.

As noted above, Fuchs does not teach or suggest processing ***a swarm of messages***, as explicitly recited in Applicants' independent claim 1. Instead, Fuchs processes messages serially, i.e., one message at a time. *See, e.g.,* fig. 14, refs. 1410-1445. Because messages are processed serially in Fuchs, there can be no teaching or suggestion of processing a swarm of messages in parallel. Indeed, such a teaching would run contrary to the serial message-processing operations taught in Fuchs. For this reason, Fuchs cannot anticipate or render obvious the Applicants' recited ***disk information-retrieval process in the file system that is carried out on the swarm of messages in parallel.***

At page 9 of the Office action, the Examiner argues that "column 5, line 65 through column 6, line 5 is one of a plurality of references by Fuchs to parallel processing." The Applicants respectfully submit that the cited passage in Fuchs is directed to concurrent execution of application processes ("parallel processing workstations, capable of executing a plurality of concurrent processes"), and is not directed to parallel processing of a swarm of messages. In other words, although multiple processes in Fuchs may execute concurrently, each of those processes is still processing its messages serially—one message at a time. As such, there is no teaching or disclosure in Fuchs that suggests that the same group (or "swarm") of messages is processed in parallel by the concurrently executing processes.

Based on the foregoing, the Applicants respectfully submit that independent claim 1, in its present form, is allowable over the cited art since Fuchs does not anticipate or render obvious the Applicants' claimed ***a consistency point at which time results of the storage system transaction entries are committed to the disk array, an initiator process that establishes a swarm of messages with respect to the storage system transaction entries and delivers the swarm to the file system***, and ***a disk information-retrieval process in the file system that is carried out on the swarm of messages in parallel.***

Because independent claims 10, 15, 20, 26, 31, 35 and 37 comprise the same or similar patentable subject matter as independent claim 1, Applicants respectfully submit that these claims are also allowable for at least the same reasons. Furthermore, claims 2-9, 11-14, 16-19, 21-25, 27-30 and 32-35 depend on allowable independent claims and are thus allowable for at least the same reasons.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

Stephen E. Kabakoff
Reg. No. 51,276
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500